

JS Level 0



HTML



HTML

На прошлом занятии мы с вами остановились на вот такой структуре, которая генерируется с помощью Emmet:

```
<> index.html > html > head > meta
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Document</title>
8  </head>
9  <body>
10  |
11 </body>
12 </html>
```

Если у вас внутри `<body></body>` размещён элемент `h1`, удалите его.



Веб-технологии

Сегодня мы познакомимся с более интересными тегами: изображениями, аудио, видео.



img

Изображения (картинки) показываются с помощью тега `img` (сокращение от `image` – изображение):

```
<img></img>
```



Закрывающий тег

На прошлом занятии, рассматривая элементы, мы говорили, что есть открывающий и закрывающий тег.

Если между тегами элемента нет содержимого, вы можете встретить сокращённую запись: `` -> ``.

Но и это не всё: для некоторых элементов можно вообще оставлять только открывающий тег: ``.



Закрывающий тег

Как определить, для каких элементов закрывающий тег нужен, а для каких - нет?

Достаточно просто:

1. Нужно открыть спецификацию HTML: <https://html.spec.whatwg.org>
2. Пролистать в оглавлении до нужного раздела (элементы будут подсвечиваться оранжевым цветом):

4.8 Embedded content

4.8.1 The **picture** element

4.8.2 The **source** element

4.8.3 The **img** element ←

3. В карточке описания посмотреть на блок **tag omission** (отсутствие тега):

Tag omission in text/html:

No **end tag**.

Если написано "**No end tag**" (нет закрывающего тега), то закрывающий тег можно не писать.



Атрибуты

Это всё хорошо, элемент мы написали, но картинки-то нет? Чтобы задать саму картинку, нам понадобятся атрибуты.

Атрибуты – это "свойства" элемента (примерно так же, как у объектов в реальной жизни есть свойства, например цвет у автомобиля, у HTML элементов есть свои свойства).



Атрибуты

Атрибуты задаются в открывающем теге в виде: `названиеАтрибута="значениеАтрибута"`.

Так же, как и для тегов – есть сокращения. Если значение атрибута – пустая строка (т.е. `названиеАтрибута=""`, то допускается не писать `=""`, т.е. достаточно написать лишь `названиеАтрибута`).

Обратите внимание: в HTML принято использовать двойные (для этого надо нажать на `Shift` и на тот же символ кавычки).



Атрибуты

Например, для элемента `img` есть атрибут `src` (source – источник), который позволяет указать адрес картинки в сети Интернет.

Обязательно пользуемся Emmet, пишем название тега элемента и нажимаем `Tab`:

```
<body>  
  img  
</body>  
</html>
```

Emmet Abbreviation

- `img:s`
- `img:sizes`
- `img:srcset`
- `img:z`

VS Code сам поставит курсор внутри скобок и даст нам написать значение первого атрибута:

```
<body>  
  <img src="" alt="">  
</body>  
</html>
```



Атрибуты

В качестве значения укажем следующее: <https://alif-skills.pro/media/logo.svg> и нажмем на клавишу **Tab**, чтобы перейти к заполнению атрибута **alt**:

```
<body>  
    
</body>  
</html>
```

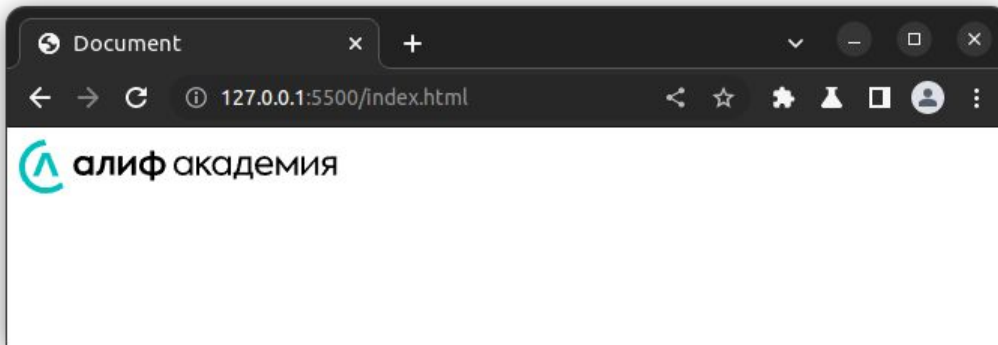
alt – это атрибут, определяющий текст, который будет показан пользователю, в случае если картинка по каким-то причинам станет недоступна (например, её случайно удалят с сервера):

```
<body>  
    
</body>  
</html>
```



Атрибуты

Теперь откроем страницу с помощью Live Server. Мы должны увидеть вот такой результат:



Т.е. действительно, браузер вставил на нашу страницу картинку, адрес которой мы ему указали в атрибуте `src`.

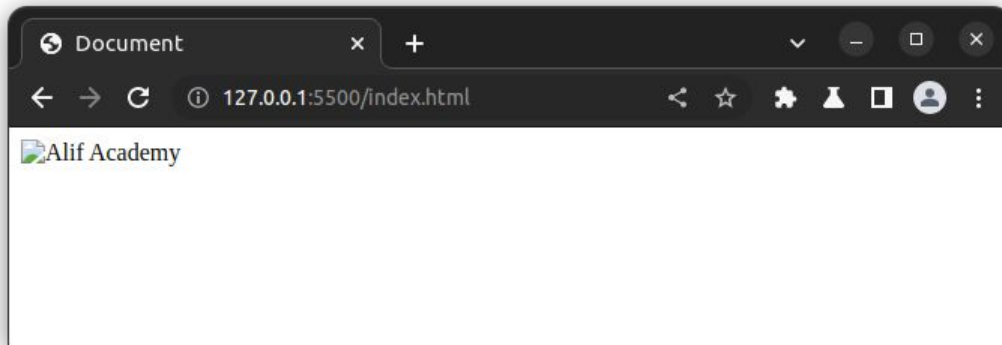



Атрибуты

Теперь "сломаем" нашу страницу, указав неправильный адрес (мы просто добавили двойку в конце адреса картинки:

```
<body>  
    
</body>
```

В браузере мы увидим следующее:



Что произошло? Браузер попытался скачать картинку, но не смог (потому что адрес неправильный). Поэтому он нарисовал иконку "битой картинки"  и отобразил текст из атрибута `alt`.

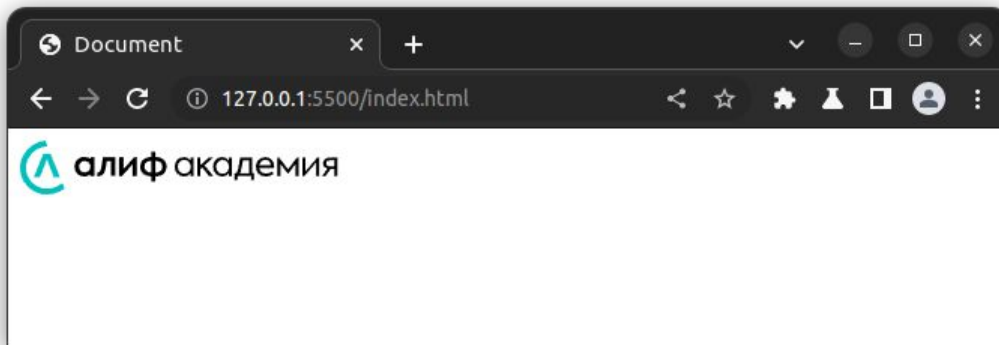


Атрибуты

Поправим адрес на верный:

```
<body>  
|   
</body>
```

Изображение снова вернётся на страницу:



Таким образом: если картинка доступна и браузер может её скачать, то он показывает картинку и не показывает текст из атрибута **alt**, если же картинка недоступна, то браузер показывает текст из атрибута **alt**.



Q & A

Q: но откуда мы узнаем адрес изображения? Почему он именно такой?

A: чаще всего мы будем знать адрес изображения исходя из того, что сами его "разместим" по этому адресу в сети Интернет (как это сделать узнаем через несколько занятий). Пока же мы будем давать вам готовые адреса.

Q: где можно узнать про все атрибуты и как они работают?

A: в спецификации HTML.



Атрибуты

Спецификация – это наш первоисточник. Всякие обучающие сайты и видео – это очень хорошо, но смотреть всегда нужно в спецификацию, потому что именно там будет наиболее полная и достоверная информация:

Content attributes:

Global attributes

alt — Replacement text for use when images are not available

src — Address of the resource

srcset — Images to use in different situations, e.g., high-resolution displays, small monitors, etc.

sizes — Image sizes for different page layouts

crossorigin — How the element handles crossorigin requests

usemap — Name of [image map](#) to use

ismap — Whether the image is a server-side image map

width — Horizontal dimension

height — Vertical dimension

referrerpolicy — [Referrer policy](#) for [fetches](#) initiated by the element

decoding — Decoding hint to use when processing this image for presentation

loading — Used when determining loading deferral



Переводчик

Но спецификация на английском языке, и если мы его не знаем, то нам приходится достаточно тяжело. Конечно же, английский язык нужно учить, но мы можем воспользоваться переводчиками, которые работают в виде сайтов.

Что нужно сделать: выделяйте левой кнопкой мыши нужный текст (кликайте левой кнопкой мыши чуть левее текста, чтобы не кликнуть на сам текст, и двигайте мышью слева-направо, выделяя текст – отпускать при этом левую кнопку не надо):

Content attributes:

[Global attributes](#)

alt — Replacement text for use when images are not available

src — Address of the resource

srcset — Images to use in different situations, e.g., high-resolution displays, small monitors, etc.

sizes — Image sizes for different page layouts



Переводчик

Content attributes:

Global attributes

alt — Replacement text for use when images are not available

src — Address of the resource

srcset — Images to use in different situations, e.g., high-resolution displays, small monitors, etc.

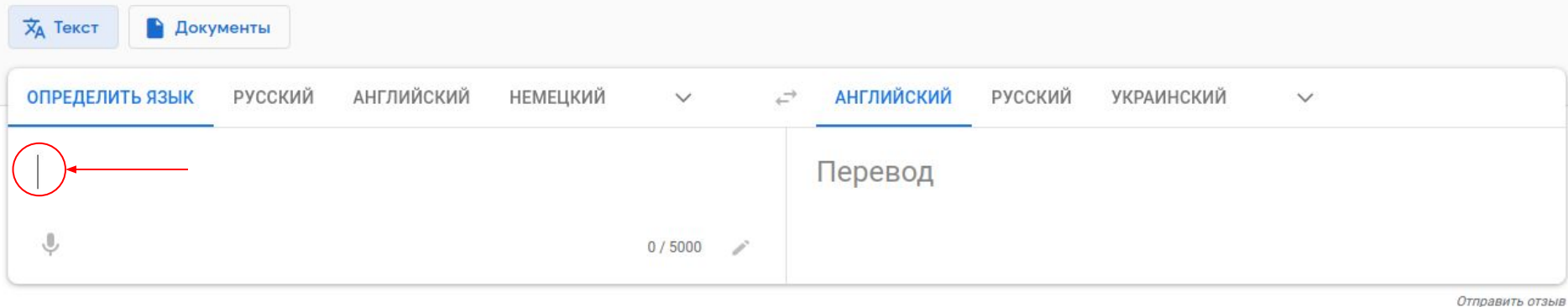
sizes — Image sizes for different page layouts

После этого (как выделили), нажимаете на клавиатуре **Ctrl + C** (скопировать текст).



Переводчик

Теперь переходим на сайт translate.google.com:



Текст Документы

ОПРЕДЕЛИТЬ ЯЗЫК РУССКИЙ АНГЛИЙСКИЙ НЕМЕЦКИЙ ▼

↔ АНГЛИЙСКИЙ РУССКИЙ УКРАИНСКИЙ ▼

Перевод

0 / 5000

Отправить отзыв

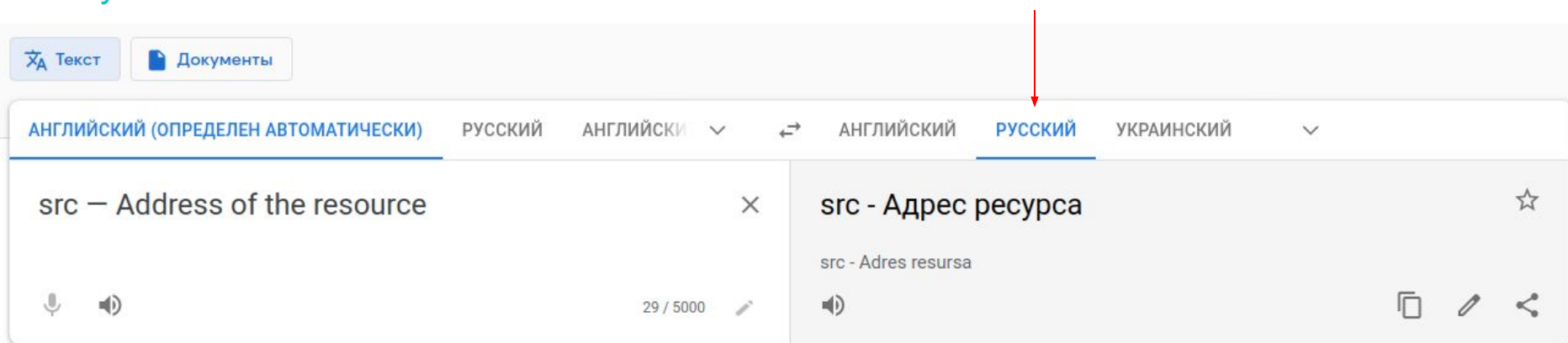
И нажмите **Ctrl + V** ("вставить" скопированный текст).



Переводчик

Затем кликните справа на нужном языке (на который нужно перевести), например,

Русский:

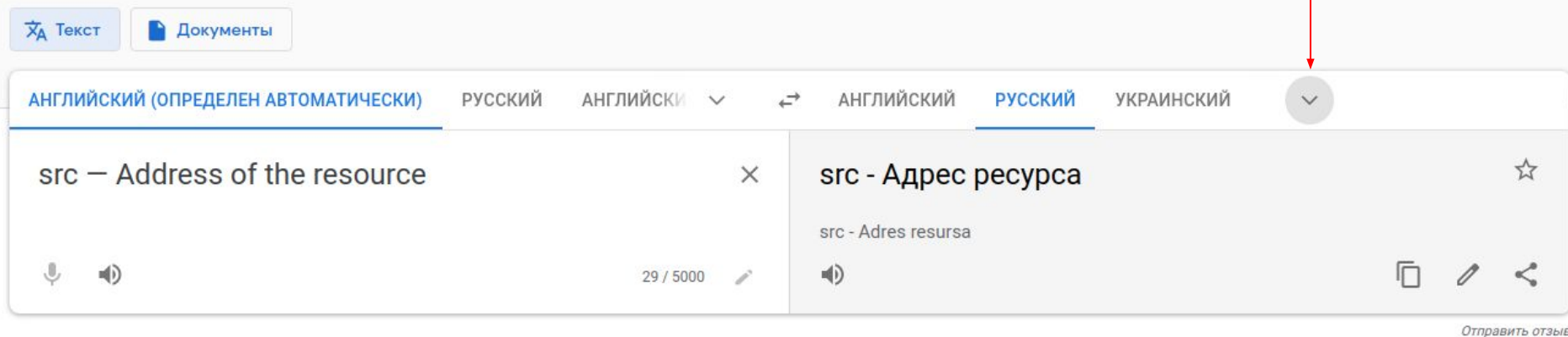


Система за вас переведёт фразу.



Переводчик

Если нужный вам язык не отображается, то вы можете выбрать из всех доступных языков, нажав на кнопку отображения списка:



← Найти языки

азербайджанский	грузинский	корейский	немецкий	словацкий	фризский
албанский	гуджарати	корсиканский	непальский	словенский	хауса
амхарский	датский	креольский (Гаити)	нидерландский	сомалийский	хинди
🕒 английский	зулу	курманджи	норвежский	суахили	хмонг
арабский	иврит	кхмерский	ория	суданский	хорватский
армянский	игбо	кхоса	панджаби	таджикский	чева
африкаанс	идиш	лаосский	персидский	тайский	чешский
баскский	индонезийский	латинский	польский	тамильский	шведский
белорусский	ирландский	латышский	португальский	татарский	шона
бенгальский	исландский	литовский	пушту	телугу	шотландский (гэльский)
бирманский	испанский	люксембургский	руанда	турецкий	эсперанто
болгарский	итальянский	македонский	румынский	туркменский	эстонский
боснийский	йоруба	малагасийский	✓ русский	узбекский	яванский
валлийский	казахский	малайский	самоанский	уйгурский	японский
венгерский	каннада	малаялам	себуанский	украинский	
вьетнамский	каталанский	мальтийский	сербский	урду	
гавайский	киргизский	маори	сесото	филиппинский	
галисийский	китайский (традиционн...	маратхи	сингальский	финский	
греческий	китайский (упрощенны...	монгольский	синдхи	французский	

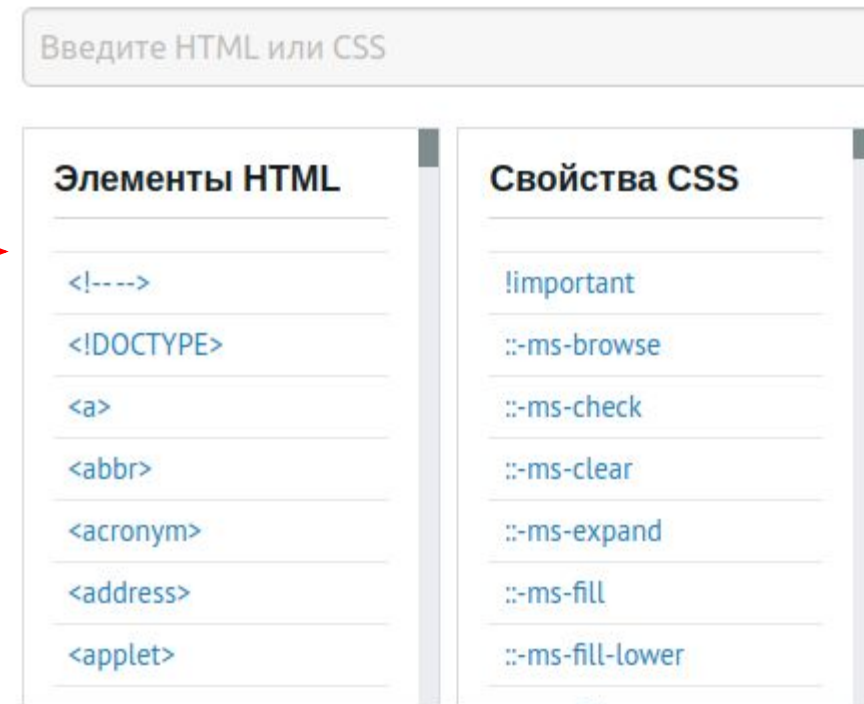


webref

Переводчик и спецификация – это, конечно, хорошо, но иногда нам захочется более простого описания (хотя лучше всё-таки привыкать сразу к спецификации).

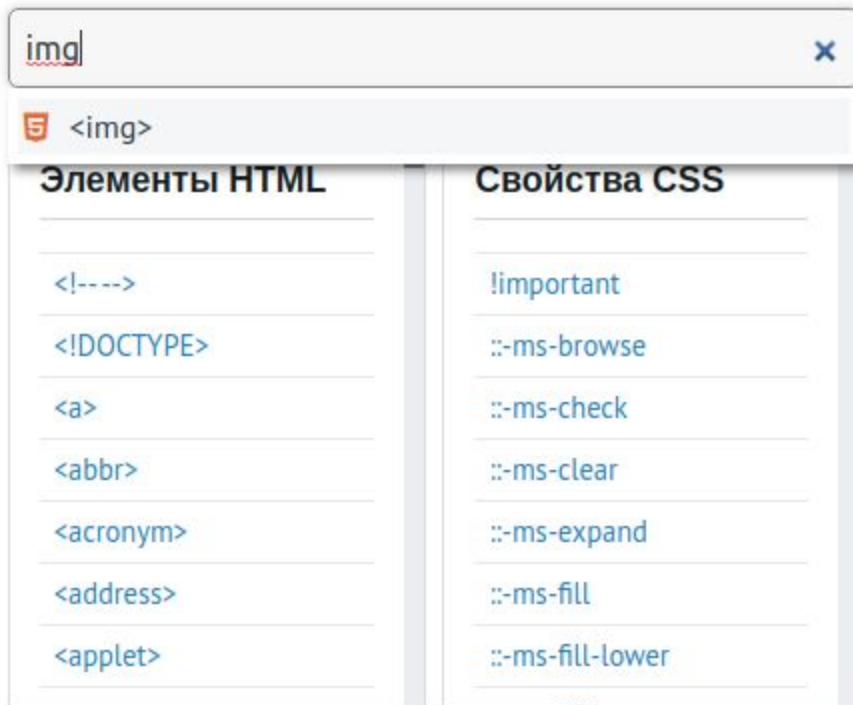
Для этого есть специальные веб-сайты, один из лучших на русском языке (на таджикском пока нет) – webref.ru.

Заходим на этот сайт и
в правом меню будет панелька
с элементами:



webref

Вводим название нужного элемента и нажимаем **Enter**:



webref

И уже на русском языке читаем описание вместе с примером использования:

Синтаксис

```
">
```

Закрывающий тег

Не требуется.

Атрибуты

<code>align</code>	Определяет, как рисунок будет выравниваться по краю и способ обтекания текстом. Устарел
<code>alt</code>	Альтернативный текст для изображения. Обязательный
<code>border</code>	Толщина рамки вокруг изображения. Устарел
<code>height</code>	Высота изображения.
<code>hspace</code>	Горизонтальный отступ от изображения до окружающего контента. Устарел
<code>ismap</code>	Говорит браузеру, что картинка является серверной картой-изображением.
<code>longdesc</code>	Указывает адрес документа, где содержится аннотация к картинке. Устарел в HTML5
<code>sizes</code>	Задаёт размеры изображения для разных макетов страницы. HTML5
<code>src</code>	Путь к графическому файлу. Обязательный
<code>srcset</code>	Путь к графическим файлам с учётом размера изображения и устройств. HTML5
<code>vspace</code>	Вертикальный отступ от изображения до окружающего контента. Устарел
<code>width</code>	Ширина изображения.
<code>usemap</code>	Ссылка на элемент <code><map></code> , содержащий координаты для клиентской карты-изображения.

Также для этого элемента доступны [универсальные атрибуты](#) и [события](#).



audio и video

С элементом для отображения изображений мы кратко познакомились, теперь давайте попробуем поработать с элементами для отображения аудио и видео.



audio

Для отображения аудио-элементов используется элемент `audio`. Так же, как и для изображений, нам необходим файл, в котором и будут содержаться сами данные, например, музыка*. Удалим элемент `img` и на его место (с помощью Emmet) поместим таким же образом элемент `audio`:

```
<body>  
|   <audio src="https://alif-skills.pro/media/ukulele.mp3"></audio>  
</body>
```

Примечание*: не бывает просто файлов изображений, аудио и видео. У большинства файлов есть владельцы, которые могут не разрешать просто так использовать их изображения, аудио и видео. Поэтому будьте внимательны, используя чужие произведения (нужно получить соответствующие права на использование). Музыку мы взяли с сайта bensound.com, который позволяет использовать их медиа-файлы, если указывается на то, что именно они – авторы.



audio

Обратите внимание: у `audio`, как и у `video`, в отличие от `img`, закрывающий тег обязателен.



audio

Если мы разместим подобным образом элемент, то не увидим ничего на страничке.

Дело в том, что элемент аудио по умолчанию никак не отображается. С помощью специального атрибута `controls` можно настроить отображение элементов управления (значение атрибута `controls` – пустая строка, поэтому можно писать не `controls=""`, а просто `controls`):

```
<body>  
| <audio src="https://alif-skills.pro/media/ukulele.mp3" controls></audio>  
</body>
```

Получим вот такой результат:



Атрибуты

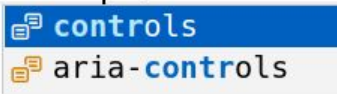
Ещё раз отдельно напомним: `controls` как раз является примером того, что мы обсуждали – вместо `controls=""` мы пишем сокращённую версию.



Атрибуты

Когда мы будем набирать код, VSCode будет помогать нам, указывая те атрибуты, которые он знает:

```
<body>  
<audio src="https://alif-skills.pro/media/ukulele.mp3" contr|></audio>  
</body>  
</html>
```



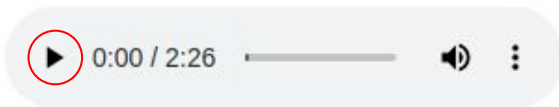
Обратите внимание: если атрибута, который мы набираем там нет, то, скорее всего, мы пишем неправильно название атрибута. Если атрибутов слишком много, то нужно просто набрать больше символов, чтобы остались нужные.

Кроме того, если атрибут подсвечен (перемещать подсветку можно с помощью клавиш вверх и вниз), то нажатие на **Tab** приведёт к тому, что VSCode за вас до конца допишет название атрибута.



audio

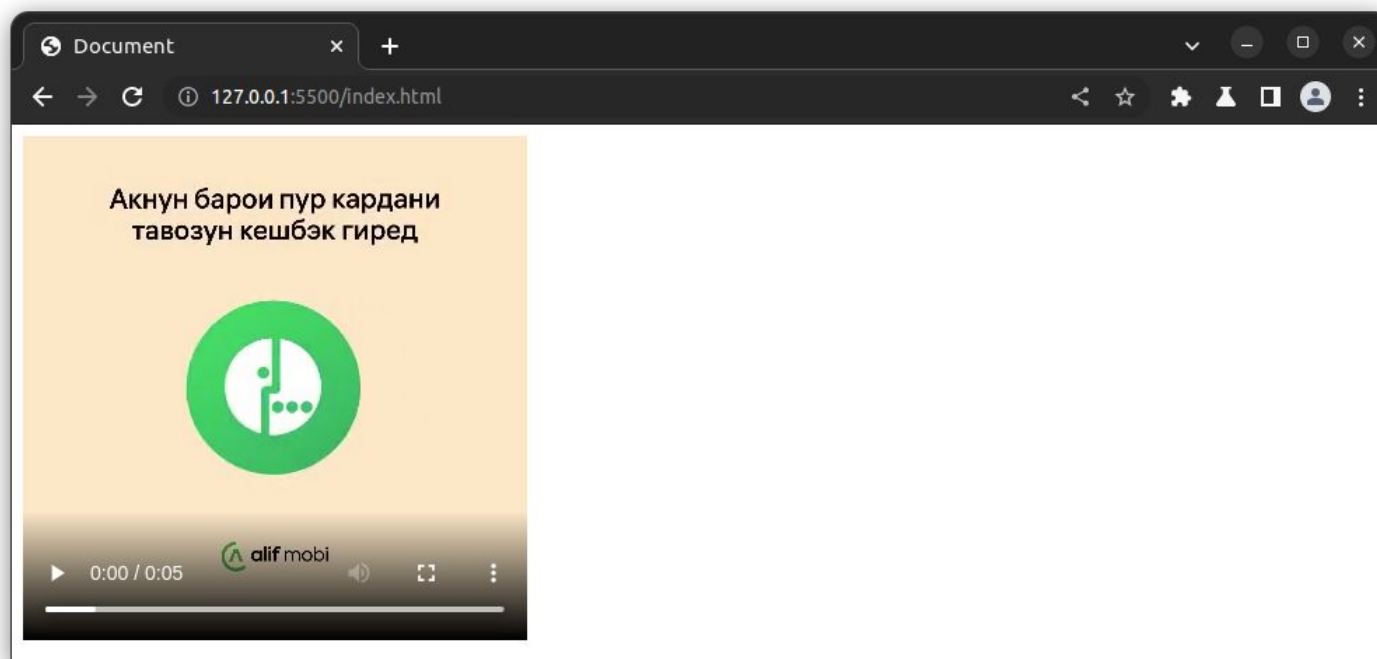
Для начала проигрывания **audio** и **video** обычно используется специальная кнопка, которая называется Play (выглядит в виде треугольника):



video

С элементом видео всё то же самое, что и с [audio](#):

```
<body>  
  <video src="https://alif-skills.pro/media/mobi.mp4" controls></video>  
</body>
```

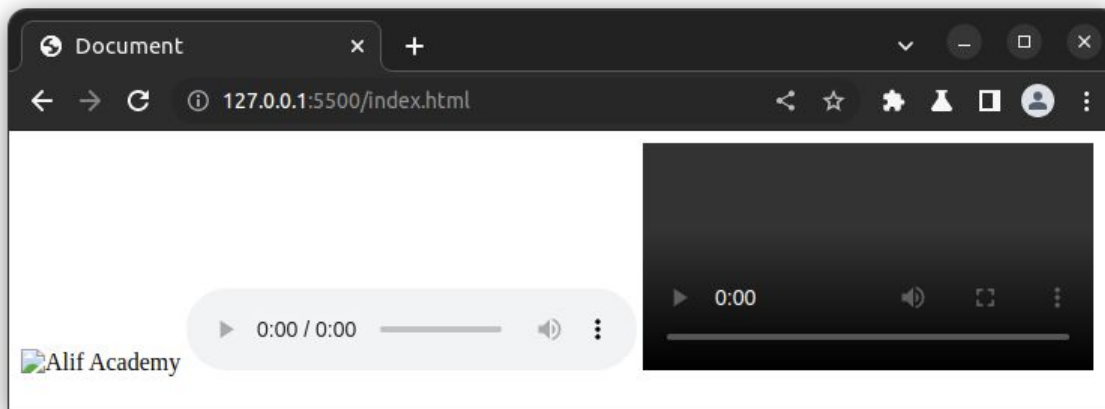


Ошибки

Что будет, если мы случайно ошибёмся и укажем неправильные адреса? Для картинок мы уже видели, теперь посмотрим всё вместе:

```
<body>  
    
  <audio src="https://alif-skills.pro/media/404.mp3" controls></audio>  
  <video src="https://alif-skills.pro/media/404.mp4" controls></video>  
</body>
```

Для изображения будет отображаться "битая" картинка, для аудио и видео – заблокированный плеер:



Q & A

Q: когда такое может произойти?

A: например, вы сделали документ, выложили в Интернет. А кто-то (например, владелец другого сайта), взял и удалил изображения со своего сайта, адреса которых вы использовали.



alt

Мы уже знаем, что у изображений (элемента `img`) предусмотрен специальный атрибут `alt` (альтернативное текстовое описание), который используется в случаях:

1. Не удалось использовать картинку
2. Пользователь использует не графический браузер (например, голосовой - он может прочитать описание картинки, но показать её не сможет).

Для аудио и видео атрибута `alt`, к сожалению, нет (придётся прибегать к помощи JS).



ИТОГИ



Итоги

В этой лекции мы начали знакомиться с новыми элементами, поговорили про спецификацию и сайт [webref](#).

Обязательно используйте полученные в практике (в том числе работу со спецификацией и сайтом [webref](#)).



ДОМАШНЕЕ ЗАДАНИЕ



Орг.моменты

Практикум состоит из 8 обязательных занятий. Мы выкладываем новые занятия каждую субботу в 12:00 (по Душанбе).

Каждую пятницу в 23:59 (по Душанбе) дедлайн сдачи домашнего задания.

Если не успеете сдать в срок домашнее задания, тогда этот практикум будет для вас закончен и вы сможете зарегистрироваться на запуск следующего через несколько месяцев.

Все вопросы вы сможете задавать в [Телеграм канале](#).



ДЗ: Картинки

Создайте проект аналогично тому, как мы это делали на лекции. Внутри элемента `body` разместите изображение (кроме изображения внутри `body` ничего быть не должно), которое находится по адресу: <https://alif-skills.pro/media/js.svg>.

В качестве альтернативного текста используйте "Неофициальный логотип JS".

Прочитайте на сайте [webref](#) (или спецификации про размеры изображения - ширину и высоту). Установите их в значения 300 (и ширину, и высоту).

Важно: архивировать также нужно каталог `docs`.

Важно: бот будет требовать в этом задании указания атрибутов ровно в том порядке, в котором они перечислены в задании. Сделано это для того, чтобы тренировать вашу внимательность (это крайне важно в программировании).



ДЗ: Видео

Создайте проект аналогично тому, как мы это делали на лекции. Внутри элемента `body` разместите видео, которое находится по адресу (кроме элемента `video` внутри `body` ничего быть не должно): <https://alif-skills.pro/media/spring.webm> с элементами управления и с атрибутом `autoplay`.

Важно: архивировать также нужно каталог `docs`.

Важно: бот будет требовать в этом задании указания атрибутов ровно в том порядке, в котором они перечислены в задании.

Важно: бот в этой задаче будет трактовать атрибуты с пустым значением `<video autoplay></video>` как `<video autoplay=""></video>`. Вас это не должно смущать, это сделано для того, чтобы учесть все сценарии. Вы можете писать без `=""`, бот примет и такой вариант.



Спасибо за внимание

alif skills

2022г.

